# Views of Software Development Work'

*Robert Edelson*
Jet Propulsion Laboratory
Pasadena, CA 91109
(818) 306-6279
redelson@devvax.jpl.nasa.gov

## ABSTRACT

Growing at about 12% per year, software development is a nearly $100 billion industry in the United States, Yet its productivity has remained almost constant. Collaborative work styles are touted as a way to improve productivity and an industry has formed about computer tools to assist and even enable collaborative work. This paper reports on a study on the attitudes of software developers regarding the "best" work styles for their tasks, the actual way in which their organizations perform them, and the innovation and skill that these tasks require. implications for Computer-Supported Cooperative Work product acceptance and leverage are discussed.

## INTRODUCTION

There is a great deal of turmoil in the world of software development over the best mix of work - styles, tools and techniques, A key issue concerns the tradeoff between teamwork and individual effort for each of the development roles. Coleman, in an address at Groupware '93 noted that "Groupware [computer tools to assist in collaborative work] was a $3.1 billion industry in 1991, and is projected to be a $10 billion industry by 1996 (hardware, software and consulting included). " (Coleman 1993), but most authors caution that the highest hurdle in introducing a collaborative work style is people's resistance to change (cf., Hsu 1993, LaPlante 1992, Norman 1991). It is important to know what views people have of desirable work styles and actual work styles for their differing kinds of work, This paper reports the results of a study to determine, for software development, what people's opinions are on best work style, actual work style, the nature of the work, and the skill required to do it for a spectrum of software development tasks. Are "best" and actual different? What is hard and what easy? Where is individual effort best and where team effort? What personal, role, and organizational characteristics affect these views?

## IMPORTANCE OF THE PROBLEM

Software development is a multi-billion dollar industry in the United States and it is an industry in which the United States has a competitive advantage (Schrage 1993, *Aviation Week 1990).* In packaged software (whose sales are expected to grow by over 12% in 1994), the United States owns 75% of the world market (Brandt 1994). Brandt estimates that software and data processing as a whole will contribute $91.4 billion to the U.S. economy in 1994, employing 2.27 million workers. Yet, industry productivity has remained constant at about $40,000 per employee (1993 dollars) since 1989, lower than any other major industry and on a par with restaurant productivity *(Business Week 1994).* For many firms, software development is practiced with much the same approach as it has been since the beginning of the industry, albeit with better tools: that is, largely by groups of individuals. Are there better ways, more productive ways, of developing software?

[1] The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, m service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States government or the Jet Propulsion Laboratory, California Institute of Technology.

In the last seven or eight years a group of increasing size has answered this question in the affirmative. Their answer is computer-supported collaborative work (CSCW) and the tools that facilitate this work are called in the developers' argot, "groupware." Claims for increased productivity through groupware are hard to substantiate, although there has been quantitative examination of the use of collaborative teams in a number of areas, particularly in the performance of reviews and other efforts at group decision support (Kirkpatrick 1992, Hamilton 1992). Review and quality are only a part of the entire puzzle and it is of interest to understand what tasks are believed *by soft ware developers* to be most amenable to collaborative work styles, with the thought that this information will assist them in targeting the most appropriate areas of their work for experiments in collaboration.

## REVIEW OF THE LITERATURE

Michael Schrage's 1990 book *Shared Minds*, is perhaps the best popular exposition on the benefits of collaboration generally and on the use of computer tools to assist collaboration. Schrage's "core theme" is

> "that people should understand that real value in the sciences, the arts, commerce, and indeed, one's personal and professional lives, comes largely from the process of collaboration. What's more, the quality and quantity of meaningful collaboration often depends upon the tools used to create it."

Reminiscent of Barnard's classic treatise where he says (Barnard 1938),

> "Formal organization is that kind of cooperation among men that is conscious, deliberate, purposeful." (p. 4)

and

> "I imitation of possibilities is necessary to choice. " (p. 14),

Schrage defines collaboration as a "purposive relationship" that must "solve a problem, create, or discover something within a set of constraints, " where the constraints include expertise, time, money, competition, and conventional wisdom.

Schrage develops a 2 x 2 taxonomy for collaboration (Figure 1). One dimension is conceptual. Tasks that require high conceptual collaboration are those that "yield insights into fundamental notions of the problem." The other dimension is technical. "Technical collaborations are the attempts to solve problems the conceptual collaborations identify. " Each task can be classified by its high or low position with respect to the need for collaboration of the two kinds. Schrage makes the intriguing comment that "Sometimes, collaborators see themselves working in different quadrants [of this matrix]." If true, such dissonant perceptions could interfere with the implementation of collaboration.

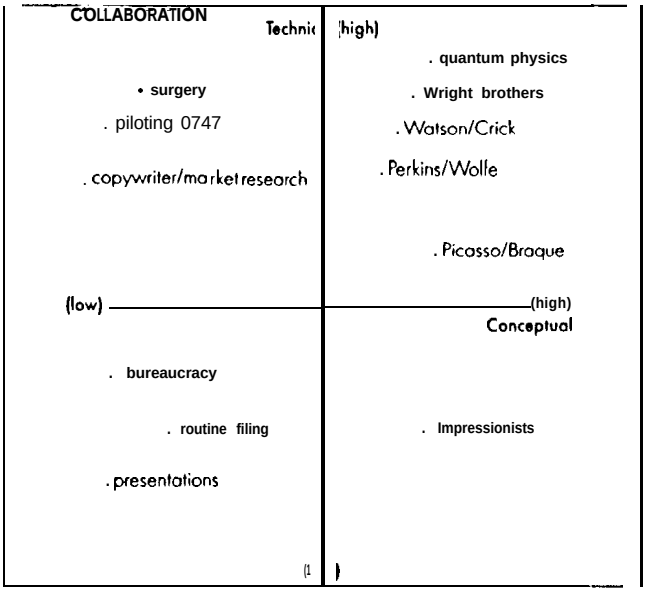



**Figure 1, Schrage's Taxonomy of Collaboration, (Schrage 199 p. 60, reproduced by permission)**

Despite Schrage's enthusiasm, others point out that adopting collaborative work styles must be done with foresight and caution. Norman (1991) makes a key point.

> "The tasks, the culture, the social structure, and the individual human arc all essential components of the job, and unless the computational tools fit 'seamlessly' within this structure, the result will be failure. "

In reporting a successful experiment in collaboration, Bly et al (1993) note that people need to be able to adapt collaborative tools to the problems being examined and to their preferred ways of working, "one size does not fit all"; the tools need to be easily accessible; a commonality of purpose and an openness about work are essential to effective collaboration. Gantt and Beise (1993) briefly relate the results of a survey given to users of a Group Decision Support System that indicated very favorable opinions on its value in reaching consensus.

CSCW is part of a wave of thought that teamwork is better than one-work. in a" much quoted *Harvard Business Review* article entitled, "Entrepreneurism reconsidered: the team as hero, " Reich (1987) says that if it was ever true that individual heros supported by worker drones created American industry, it is no longer, Increasingly we recognize that,

> "success comes through the talent, energy, and commitment of a team—through *collective* entrepreneurship.

People's *views* of collaboration, of the firm, and of management inevitably affect the implementation of collaboration. Therefore, in exploring people's opinions on, and experience with collaboration, it is important to establish something of how they think. Some people may fear collaboration and loss of individuality. Personal beliefs might strongly affect answers to questions on collaboration.

Fundamentally, the question of collaborative effectiveness is not a dilemma of inadequate tools or methodologies, it is a question of the *appropriate choice of tasks,* and this is a question that has been debated by group dynamicists for decades. Saavedra et al (1993) found support for the theories of Galbraith (1987) and March and Simon (1 958), in experiments conducted with small groups, demonstrating that collaboration is most effective for tasks that require "complex interdependence. "

> "Generally, the greater the number of dimensions that define interdependence for a work group, the more complex the interdependence and the greater the need for collaboration and mutual adjustments among group members. "

## THE RESEARCH QUESTtONS

Following Schrage, we define collaboration as a purposive relationship that must solve a problem, create, or discover something within the constraints of expertise, time, money, competition, and conventional wisdom. Distinguishing *conceptual* collaborations as those that yield insights into fundamental notions of the problem and *technical* collaborations as those that solve problems that conceptual collaborations identify, to what extent is collaboration practiced by software developers and their managers? Do the perceptions of managers and developers differ on the nature of the collaborative task and on the extent to which collaboration is used? Can we characterize the extent of collaboration on Schrage's technical and conceptual axes?

"Extent," "conceptual, " and "technical" are difficult concepts to quantify and examine. We have attempted to operationalize the above questions by formulating the research questions of this study as follows:

*Software development is comprised of a number of distinct tasks which the people involved perform to various degrees depending on their primary roles. According to these people, what level of collaboration is practiced in each of these tasks ? What level would be desirable? Do people with different primary roles have different perspectives as to the creativity and skill required by these tasks?*

## METHODS

A list of *200* people involved in computer work was developed by systematic random sampling from the *1993 IEEE (Institute of Electrical and Electronic Engineers) Membership Directory* list of Affiliates of the Computer Society. Eliminated in developing the list were those people for whom it was clear by title that the individual is not involved in software development (e.g., Senior Financial Vice President), is not in the USA, is part of a university, or is an acquaintance of the author. A questionnaire and cover letter guaranteeing confidentiality were mailed to each of the identified people. Questionnaires that proved undeliverable and questionnaires which the recipients returned, but did not answer were dropped from the sample, Ultimately, 71 completed questionnaires were received out of a sample of 169 for a response rate of 42%.

One of the questions that this study seeks to answer is "What tasks do people see as appropriate for collaborative work rather than individual work." Schrage's provocative suggestion that collaborators may have different views of work on the axes of conceptual and technical work characterization was operationalized by questioning recipients on best and actual work styles for various software development tasks and comparing managers' with non-managers' views. The "distinct tasks" of software development were partitioned into nineteen categories, each of which is generally associated with a development role as shown in Table 1. "Level of collaboration" is defined by a five point Likert scale with categories: (1) Individual Effort; (2) Some Consultation with Others; (3) Wide Consultation with Others; (4) Team Advice and Review; and (5) Team Product. This approach separated out the individual versus teamwork aspects of collaboration. In the case of "actual" work style, a sixth category was added: "Not done here."

For the same task categories, the recipients were asked to rate: (1) the "nature" of the task; and (2) the skill required to do it. The nature of the work, which we equate with "creativity" is quantized in a five point scale ranging from (1) Routine to (5) Innovative. "Level of Skill" is defined by the need for: (1) Novice; (2) General Competence; (3) Excellence; (4) Expert; and (5) World Class talents to achieve satisfactory performance. "Nature" thus takes up the question of conceptuality while skill addresses the technical dimension, and the combination with views. of appropriate levels of teamwork gets at Schrage's conceptual and technical dimensions.

## DESCRIPTIVE STATISTICS

Space limitations prevent displaying all of the data or even a copy of the questionnaire, however, Table 2 presents frequency distributions of the data on "best" and actual work styles, the heart of this particular study. Instances of no response are not included in percentages, means or standard deviations. Other data are summarized below. (A more lengthy discussion is available upon request from the author.) for variables of particular interest. In this section, we discuss some of the more interesting results from examination of the frequency distributions and the results of comparisons between groups of respondents.

### Sample Characterization

As no effort was made to equalize the numbers of men and women responding, the respondents were heavily male (84.5%) in approximate agreement with the à priori percentage of men and women in the original mailing. The respondents were mature, experienced people with an average age of 42 years (median, 41) and an average of 14.5 years (median, 13) in software development, They reported an average of 9.8 years (median 8) in the specific primary role in which they now act. Although 83% of the respondents named a specific primary role, 98.5% of these routinely performed in other capacities; the median was four to six roles of the seven specified (see

Table 1). This was also a highly educated group, with a mean of 6.5 years (median 6) of education beyond high school,

On a personal level, the respondents were neither introverts nor extroverts, averaging just about neutral on questions such as "I view myself as an introvert" and "I need a great deal of privacy. " There was a bent toward regarding talking with coworkers as important and most have had highly collaborative work experience. A very strong agreement was shown with one aspect of their personal situation: 91.4% either agreed or strongly agreed that "I understand the goals and objectives of my task assignments. " To those who fee] that American workers are alienated from their workplace, this suggests that software developers are not among that group.

Table 1. Job Roles and Task Categories

| Job Roles | Task Categories |
|---|---|
| Management | Planning, Progress Analysis, Quality Measurement, Work Organization, Process Design, Review |
| Systems Engineering | Tradeoffs, Requirements Development, Requirements Analysis, Requirements Partitioning, Development Environment and Tool Selection |
| Software Development and/or Maintenance | Prototyping, Software Design, Coding, Software Integration |
| Testing | Testing, Failure Reporting |
| Documentation | Documentation |
| Training | Training |

Table 2. Frequency Distributions, Best and Actual Work Styles

| Task | Best and Actual Work Styles (% of respondents) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Individual effort 1 | Some consultation with others 2 | Wide consultation with others 3 | Team advice and review 4 | Team product 5 | Not done here 6 | Mean | Standard deviation |
| Planning, Best/Actual | 4.2/7.0 | 14.1/60.6 | 31.0/ 11.3 | 25.4 / 11.3 | 25.4 /5.6 | —/4.2 | 3..%/2,61 | 1.14/1.21 |
| Progress analysis, Best/Actual | 8.7/24.6 | '29.0 /39.1 | 10.1/ 10.1 | 36.2/10.1 | 15.9/8.7 | —/7.2 | 3.22/2.61 | 1.27/1.54 |
| Quality measurement, Best /Actual | 7.4/24.3 | 25.0 /30.0 | 13.2/ 10.0 | 25.0/ 7.1 | 29.4/ 4,3 | —/24.3 | 3.44/3.10 | .34 / 1.94 |
| Work organization, Best/Actual | 13.0/38,6 | 23.2/41.4 | 17.4 /5,7 | 27.5/ 8.6 | 18.8/ 1.4 | —/4.3 | 3.16/2,06 | .34/ 1.27 |
| Tradeoffs, Best/Actual | 4.5/27.3 | 18.2/36.4 | 24.2/ 13.6 | 33.3/9.1 | 19.7/9.1 | -/4.5 | 3.46/2,50 | .14/1 .45 |
| Requirements development, Best/Actual | 1.4/14.3 | 17.4 /48.6 | 30.4 /20.0 | 21.7/5.7 | 29.0/ 7.1 | —/4.3 | 3.59/2,56 | .13/1.27 |
| Requirements analysis, Best/ Actual | 2.9/26.1 | 18.8/52.2 | 26.1 /10.1 | 30.4/5.8 | 21.7/ 4.3 | —/1.4 | 3,49/2.15 | 1.12/1.10 |
| Requirements partitioning, Best/Actual | 6.0/ 24.6 | 20,9/ .43,5 | 19.4/11.6 | 37.3 /2.9 | 16.4/8.7 | —/8.7 | 3.371 ?..54 | 1.17/1.55 |
| Development environment & tool selection, Best/Actual | 8.6/24.3 | 18.6/443 | 18.6/ 10.0 | 35.7/8.6 | 18,6/ 11.4 | —/1.4 | 3,37/2,43 | 1.23/1.34 |

l'able 2. Frequency Distributions, Best and Actual Work Styles

| Task | Best and Actual Work Styles (% of respondents) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Individual effort 1 | Some consultation with others 2 | Wide consultation with others 3 | Team advice and review 4 | Team product 5 | Not done here 6 | Mean | Standard deviation |
| Review, Best /Actual | 1.4/14.3 | 14.3/40.0 | 12.9/14.3 | 42.9/ 17.1 | 28.6/ 7.1 | —/7.1 | 3.83/2.84 | 1.05/1.44 |
| ['recess design, Best/Actual | 6.2/30,3 | 26.2/37,9 | 21.5/ 4,5 | 26.2/13.6 | 20.0/ 7.6 | —/6.1 | 3.2$/2.49 | 1.23/1.53 |
| Prototyping, Best/Actual | 18.6/37.7 | 41.4 /37.7 | 10.0/ 2.9 | 20.0 /7.2 | 10,0/ 2,9 | -/11.6 | 2.61 /2.35 | 1.2$/1.65 |
| Software design, Best / Actual | 14.5/42.0 | 23. ?/36.2 | 10.1 /7.2 | 34.8/8.7 | 17.4/ 2.9 | —/2.9 | 3.17/2.03 | 1.36/1.26 |
| Coding, Best/Actual | 43.5/68.6 | 30.4/ 21.4 | 2.9/2.9 | 13.0/ 4.3 | 10.1 /1.4 | —/1.4 | 2.16/1.53 | 1.38/1.02 |
| Software integration, Best/Actual | 7.2/31.4 | 23.2/41.4 | 17.4/7.1 | 23.2/11.4 | 29,0/ 5,7 | —/2,9 | 3.44/2.27 | 1.32/1,33 |
| Testing, Best/ Actual | 17.4/37.1 | 12,7/34,3 | 17.4/ 12,9 | 17.4/ 7.1 | 34.8/7.1 | —/1.4 | 3.39/2,17 | 1.51/1.29 |
| Failure reporting, Best/ Actual | 17.9/44.9 | 29,9/ 27,5 | 13.4/7.2 | 19.4 /2.9 | 19.4 /4.3 | —/13.0 | 2.93/2.33 | 1.42/1.75 |
| Documentation, Best/Actudl | 7,2/35,7 | 30,4 /38,6 | 15.9/7.1 | 24.615.7 | 21,7/ 4.3 | —/8.6 | 3,23/2.30 | 1.30/1.54 |
| Training, Best/ Actual | 12.1 /27.9 | 24.2 /33.8 | 18.2/16.2 | 24.2/ 2,9 | 21.2/2.9 | —/16.2 | 3.18/2.68 | 1.35/1.73 |

Generally people reported modest or moderate sizes of "local work group. " The median was 12 people and 67.8'% reported less than 20 individuals in their "local work group. " On other organizationally directed questions, the averages showed a tendency toward neutral answers (neither agreement nor disagreement). For example, the questions "Generally, my coworkers and I are rewarded for individual effort, " and "Generally, my coworkers and I are rewarded for our team efforts, " had virtually identical means (neutral) and standard deviations. On this across-the-board level, the strongest tendencies were: to disagree with the statement "My local organizational environment could be considered authoritarian"; and to agree with

- "I believe that my coworkers and I have shared goals and objectives,"
- "Generally, close collaboration with my coworkers is expected,"
- "Creativity is expected in my local area,"
- "There is a climate of respect in my local area,"
- "There is a climate of trust in my local area,"
- "I frequently talk with my coworkers."
- "I understand what I am supposed to accomplish,"
- "1 understand the extent to which I have autonomy in accomplishing my task, "

and

"That autonomy is satisfactory."

On the whole then, these organizational environments seem to lean toward promoting collaborative activity while still giving individuals room to express their individuality. One respondent described such a situation:

> "Good software development requires a combination of the best individual *and* team effort. We are constantly going back and forth between hard thinking (individually) about a problem and team discussion of the alternatives. Individual efforts arc validated by the team. Team discussion feeds into individual efforts."

The trick of course is, as another participant said:

> "Complex software systems cannot be built without strong team activity. The difficulty is finding the balance between capitalizing on individual creativity and strengths while maintaining good team work."

## Work Styles

One-tailed, paired t-tests were performed on each of the nineteen tasks identified in our software development taxonomy to compare peoples' opinions of the "best" way to do each task with the "actual" way that it is done. In every case the "best" way involved significantly (<.001) more teamwork than the actual, even in the quintessence of individuality: coding. Table 3 shows the means and standard deviations for "best" and "actual" for each task.

All of the tasks had moderate to strong correlations between Nature and Skill at significance levels of ≤0.001. These correlations are shown in Table 4.

Table 3. Means and Standard Deviations for "Best" and "Actual" Task Work Styles

| Task | Means/Standard Deviations |
|---|---|
| Planning, Best | 3s4/1 .14 |
| Actual | 2.46/1.00 |
| Progress analysis, Best | 3.23/1 .26 |
| Actual | 2..34/1.25 |
| Quality measurement, Best | 3.49/1.31 |
| Actual | 2.17/1.16 |
| Work organization, Best | 3.18/1,35 |
| Actual | 1.88/0.98 |
| Tradeoffs, Best | 3.48/1 .15 |
| Actual | 2.33/1.26 |
| Requirements development, Best | 3.58/1,1 I |
| Actual | 2.39/1.07 |
| Requirements analysis, Best | 3.47/1.11 |
| Actual | 2.09/1.00 |
| Requirements partitioning, Best | 3.40/1.13 |
| Actual | 2.21/1.17 |
| Development environment & tool selection, Best | 3.39/1.23 |
| Actual | 2.38/1.27 |

Table 3. Means and Standard Deviations for "[lest" and "Actual" Task Work Styles

| Task | MeansStandard Deviations |
|---|---|
| Review, Best | 3.77/1,06 |
| Actual | 2,60/1,18 |
| Process design, Best | 3.2811.2S |
| Actual | 2.28/1.28 |
| Prototy ping, Best | 2. S4/1.27 |
| Actual | 1.87/1.04 |
| software design, Best | 3.21/1.37 |
| Actual | 1.91/1.07 |
| Coding, Best | 2.16/1.38 |
| Actual | 1.46/0.87 |
| Software integration, Best | 3.41)1,32 |
| Actual | 2.16/1,1[1 |
| Testing, Best | 3.43/1.49 |
| Actual | 213/1.21 |
| Failure reporting, Best | 2.73/1.36 |
| Actual | 1.78/1.08 |
| Documentation, Best | 3.13/1,28 |
| Actual | 1.95/1.08 |
| Training, Best | 3.05/1.38 |
| Actual | 2.05/1.00 |

These correlations may be to some degree artifacts of the questionnaire structure, which placed the two answers side by side, encouraging, perhaps, parallel answers. On the other hand, it seems reasonable to believe that the nature of the task and the skill required to perform it are related.

t-tests were also run to determine if people with different primary roles had different views on the degree of teamwork, best and actual, and on the nature and skill required for each task. Recall that Schrage (1990) hypothesized that collaborators might view technical and conceptual requirements of tasks differently and we chose to examine this

hypothesis for managers and non-managers. In fact, when managers were compared to all others on the teamwork dimension, on only one task assessment was there any significant difference, Managers felt that the best way to perform software integration requires more teamwork than the other groups declared (significance = 0.02).

Table 4. Correlations Between the Nature of the Task and the Skill Required

| Task | Correlation |
|---|---|
| Planning | 0.542 |
| Progress analysis | 0.559 |
| Quality measurement | 0.496 |
| Work organization | 0.530 |
| Tradeoffs | 0.550 |
| Requirements development | 0.705 |
| Requirements analysis | 0.689 |
| Requirements partitioning | 0.639 |
| Development environment & tool selection | 0.436 |
| Review | 0.386 |
| Process design | 0.603 |
| Prototyping | 0.549 |
| Software design | 0.498 |

Regarding comparisons of nature and skill, there were a number perceptual differences. The managers as a group rated twelve of the nineteen tasks listed as requiring more innovation or skill than did people with other primary roles. Eight of those differences are on a "conceptual" dimension (the nature of the task). Seven differ on a "technical" dimension (the skill required). On only three of the nineteen tasks were both nature *and* skill required seen to be higher: Progress analysis, Work organization, and Requirements development. Figure 2 plots a "Schrage matrix" for these tasks, where (3,3) is taken as the origin, the means for the two groups are plotted where their difference is significant and the total group mean is used where it is not ,

Figures 3 through 6 give a picture of the composite opinions of Managers, System Engineers, Others, and the combined Software Developers, Maintainers, and Testers, The points plotted are the means for each named group of the number of answers given for all questions in each category. For example, Managers answered an average of 1.47 questions as "Individual effort best" of the nineteen possible. Note the very different profiles of "best" and "actual." For "best," (Figure 3) all groups but system engineers display a bimodal distribution that prefers "Some consultation" or "Team review" to the extremes or to the middle ground. System engineers, on the other hand are biased strongly toward the "Team product"—not surprising perhaps, given the nature of their work. All of the groups saw the actual work styles being used as heavily biased toward individual efforts (Figure 4).

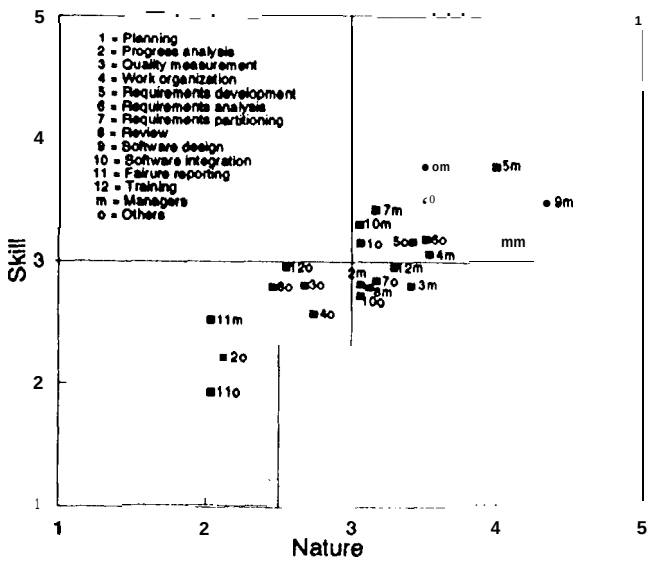
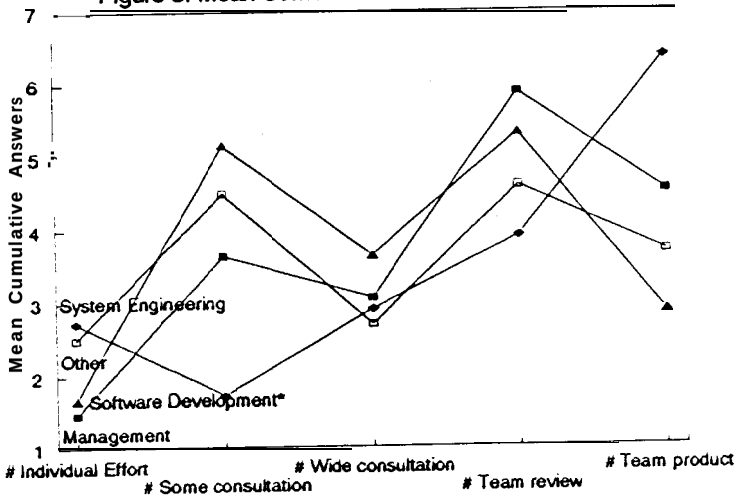


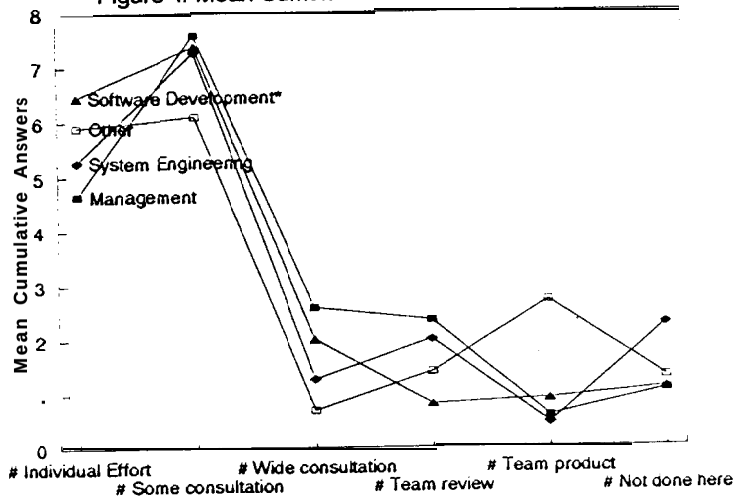

**Figure 2. A "Schrage Matrix' for Group Perceptual Differences of Task Nature and Skill Required**

Figure 3. Mean Cumulative Answers of "Best" Workstyle



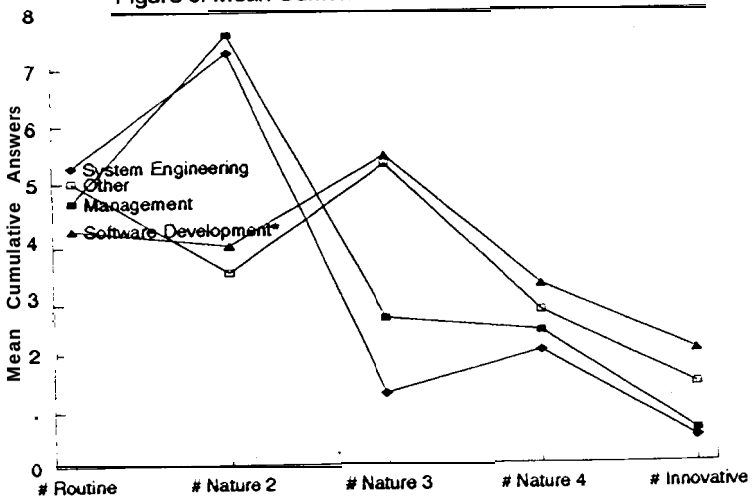

* Development + Maintenance + Testing

Figure 4. Mean Cumulative Answers for Actual Workstyle



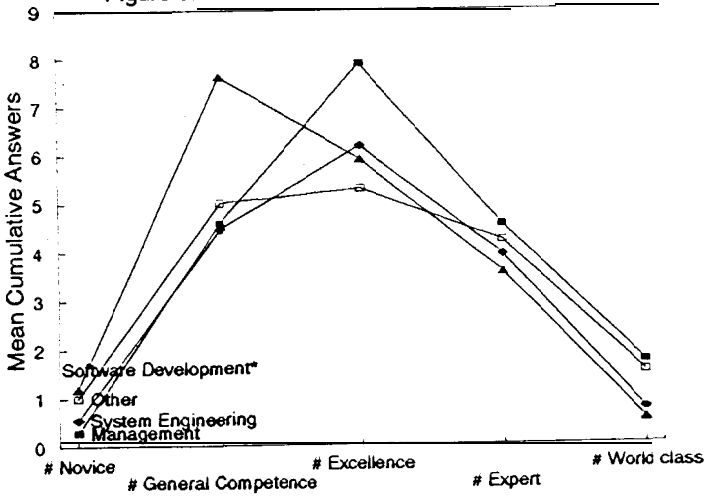

* Development + Maintenance + Testing

Figure 5. Mean Cumulative Answers for Task Nature




* Development + Maintenance + Testing

Figure 6. Mean Cumulative Answers for Skill Required




* Deve + Mai + Testing

System engineers and Managers (Figure 5) leaned toward evaluating software development tasks as routine, while Software developers and "Others" (largely consultants) had a more uniform evaluation. Software developers' opinions of skill required peaked at General Competence, while the other groups peaked (less sharply) at Excellence.

The bimodality phenomenon in "best" is quite visible in Table 2, where several of the individual variables have a pronounced bimodality in their frequency distributions, For example, the (distributions for the "best" ways of doing Progress analysis, Quality measurement, and Work organization have peaks in Some' consultation with others and Team advice and review and troughs in Wide consultation with others. The same phenomenon does not appear in the "actual" distributions, which tend to be monotonic. The conclusion we draw is that evaluations of actual work are valid descriptions of what is going on, while expressions of what is "best" appear in two camps: one that favors more individual approaches and one that prefers more team-oriented functioning.

In considering the actual work styles of their organizations, each of the tasks was not performed at one or more. Some of these reports (e.g., a report that coding was not done) may be related to what the respondent defined as his local work group. This may explain why the Quality measurement "Not done here" is 24.3%-the traditional software development model uses external quality assurance, Still . . . Other high "Not done heres" were: Requirements partitioning (8.7%); Prototyping (11,6%); Failure reporting (13%); Documentation (8.6%); and Training (16.2%). Each percentage is an interesting comment on the state of practice. Perhaps there is some truth to one received comment:

> "1 feel that the team approach is much more needed than is practiced. Too many tasks arc done by individuals without proper management. Management is the furthest behind in development in the software development cycle. "

In all this, the good news for groupware purveyors is that people feel that there is a need for more teamwork in every area of the discipline.

In constructing the questionnaire, certain relationships were hypothesized between respondents' opinions on the tasks, their personal characteristics and history, and the character of the organizations for which they work. These were:

1. People who arc introverts, who greatly value privacy, would view the "best" way to perform the tasks lying toward the "Individual effort" side of the sped rum.

2. People with greater experience in software development and in their particular job role, people with a greater breadth of experience, and people with more education would be biased toward the "Team product" approach.

3. The size of a respondent's local work group would influence his or hcr opinion on the best approach.

4. People working in authoritarian organizations or organizations which tend to dampen teamwork (through their reward systems or individualistic cultures) would lean toward "Individual effort" as best, while people working in organizations that possess collaborative environments would lean toward "Team product ."

Generally, these distinctions were not supported *as, virtually all* of the respondents felt that the best way in which to do's task is to do it in a more collaborative way than it is actually being done.

## DISCUSSION

We have asked and answered several questions in the course of this paper. Two important ones require further discussion.

1. *What tasks are believed by software developers to be most amenable to collaborative work styles ?*

Considering the opinions of the "best" work styles of the sample as a whole, the four most collaborative tasks should be: Review, Requirements development; Planning; and Requirements analysis. The latter three are also among those tasks requiring the most innovation and skill. Existing groupware tools can facilitate review. Johnson et al (1993), report an experimental tool designed specifically for review that includes automatic productivity measures and documentation. Groupware tools specifically designed for requirements development and analysis and general planning seem to have a relatively receptive user community waiting.

Correspondingly, tasks felt least amenable to collaborative work were: Coding, Prototyping; and Failure reporting.

We note that planning and requirements activities are à priori among the most "complex interdependent" tasks in software development, Their identification as targets for increased collaboration, in agreement with the experiments of Saavedra et al (1993), provides some confirmation of the external validity of the survey, as do the less interdependent skills of Coding and Failure reporting.

2. *Do people with different primary roles have different perspectives as to the creativity and skill required by these tasks?*

The answer is "yes" for the majority of the tasks and, interestingly, managers think better of the talents needed then do non-managers. The differences are generally not as great as Schrage's "different quadrant," indicating congruence in people's views of the job.

Finally, the size of the local work group did not seem to affect the respondents' responses, although one warned:

> "It has been my personal experience that teams of three or less work well with each member performing a lot of individual development. 1 am constantly amazed at how little is accomplished by each member of a large (more than 3) team, due to increased comm unication time, discussion, arguments, paperwork, hierarchy, etc. "

however, another respondent cautioned: "social characteristics are one of the least considered evaluation criteria for hiring, evaluation and promotion, " an important point for managers to consider as they promote collaboration.

## ACKNOWLEDGEMENTS

This paper would not have been written without the encouragement and assistance of Professor David Drew and Ms. Lisa Wolfe of The Claremont Graduate School. I also thank O. Demartino, J. Diehl, F. Akers, and G. Stephan of the Jet Propulsion Laboratory for their critique of the questionnaire.

## REFERENCES

*A viation Week & Space Technology, (1990).* 12 February, 95.

Barnard, Chester I., (1938). *The Functions of the Executive,* Cambridge, MA: Harvard University Press.

Brandt, Richard (1994). "Software will Play Hardball Again," *Business Week, 3353,* January 10, 82.

Bly, Sara A., Harrison, Steve R. and Irwin, Susan (1993). "Media Spaces: bringing people together in a video, audio, and computing environ merit," *Communications of the ACM, 36, 1, 28-46.*

*Business Week (1994). .?.?53,* January 10, 71-108.

Coleman, David (1993). "Welcome to GroupWare '93," *Group Ware '93 Proceedings,* San Mateo, CA: Morgan Kaufman, xi-xiii.

Galbraith, J.R. (1987). "Organization Design," in J. Lorsch (Ed.), *Handbook of organizational Behavior,* Englewood Cliffs, New Jersey: Prentice Hall, 342-357.

Gantt, James D. and Beise, Catherine M. (1993). "The Public Reacts to GDSS," *Byte, 18, 3, 120.*

Hamilton, Rosemary (1992). "Texaco finds rewards, challenges in groupware: uses workgroup approach to tackle integration," *Computerworld, 26, 36,* September 7, W.

Hsu, Jeffery and Lockwood, Tony (1993). "Collaborative Computing: Computer-aided teamwork will change your office culture forever," *Byte, 18,* ?, 113-120.

The Institute of Electrical and Electronics Engineers, inc. (1993), 1993 *IEEE Membership Directory, New* York, Volume 11, 2239-2440.

Johnson, Philip M., Danu Tjahjono, Dadong Wan, and Robert S. Brewer (1993). "Experiences with CSRS: An Instrumented Software Review Environ merit," *Proceedings of the Eleventh Pacific Northwest Soft ware Quality Conference,* Portland, OR, 301-316.

Kirkpatrick, David (1992). "1 Iere comes the payoff from PCS," *Fortune, 125, 6,* March 23, 93.

LaPlante, Alice (1992). "Group(ware) therapy," *Computerworld, 26, 30, 71-3.*

March, J. G. and Simon, 11. A. (1958). *organizations. New* York: Wiley.

Norman, Donald A. (1991). "Collaborative Computing: Collaboration First, Computing Second," *Communications of the ACM, 34, 12, 88-90.*

Reich, Robert B. (1987) "Entrepreneurism reconsidered: the team as hero, " *Harvard Business Review,* 65, 3, 77-83.

Saavedra, Richard, Earley, P. Christopher and Dyne, Linn Van (1993). "Complex Interdependence in Task-Performing Groups," *Journal of Applied Psychology, 78, 1, 61-72.*

Schrage, Michael (1990). *Shared Minds: the new technologies of collaboration. New* York: Random House.

Schrage, Michael (1993). "Software Powerhouses Remain Elusive Goal for Japanese," *Los Angeles Times,* May 13, D1.